Autonomous Robot System for Sensor Characterization

David Bruemmer, Douglas Few, Miles Walton, Heather Hunting

Idaho National Engineering and Environmental Laboratory
{bruedj, fewda, mwalton, hunthl}@inel.gov

Simon Sirin

Remote Sensing Laboratory
SirinSC@nv.doe.gov

Abstract – *This paper discusses an innovative application of new Markov localization techniques that combat the problem of odometry drift, allowing a novel control architecture developed at the Idaho National Engineering and Environmental Laboratory (INEEL) to be utilized within a sensor characterization facility developed at the Remote Sensing Laboratory (RSL) in Nevada. The new robotic capability provided by the INEEL will allow RSL to test and evaluate a wide variety of sensors including radiation detection systems, machine vision systems, and sensors that can detect and track heat sources (e.g. human bodies, machines, chemical plumes). By accurately moving a target at varying speeds along designated paths, the robotic solution allows the detection abilities of a wide variety of sensors to be recorded and analyzed.*

I. INTRODUCTION

As the number and sophistication of available sensors has increased, so has the need to accurately characterize the benefits and limitations of each sensor. Accurate sensor characterization requires precise validation of the sensor's operational performance. This is often performed by moving a target source at varying speeds along a designated path in order to accurately record and analyze sensor responsiveness. Such testing requires accurate target position information to ensure representative results. State of the practice methods require engineers to build custom tracks along which the target can be moved for each characterization experiment. One drawback of this approach is that it significantly limits the kinds of paths that can be created. Additionally, it is costly and tedious to rebuild the tracks for each experiment.

To address these limitations, the Remote Sensing Laboratory (RSL) and the Idaho National Environmental and Engineering Laboratory (INEEL) are developing an automated sensor characterization facility that utilizes a fully autonomous robot system to control target motion. With a custom graphical interface tool, the user can direct the automated capabilities of this robot in order to accomplish sensor characterization objectives. User tasking includes the ability to specify complex paths and assign velocity and acceleration profiles to each path segment. The RSL – Sensor Lab consists of a realistic and reconfigurable environment allowing the creation of test areas with varying shapes and sizes. This wide degree of adaptability allows the creation of controlled testing environments that can be used to characterize a wide variety of sensor systems. This characterization requires "sufficient comparable data" – data can only be obtained if sensors and systems are gathered and tested under the same conditions. Test repeatability is therefore of paramount importance.

Clearly a mobile robotic solution has the potential to provide immense dividends in terms of flexibility and repeatability as well as time and cost savings. However, several significant challenges must be overcome including:

- Route Specification – The ability for the user to generate complex, reusable routes, specify acceleration / velocity profiles and assess the safety and feasibility of each route.

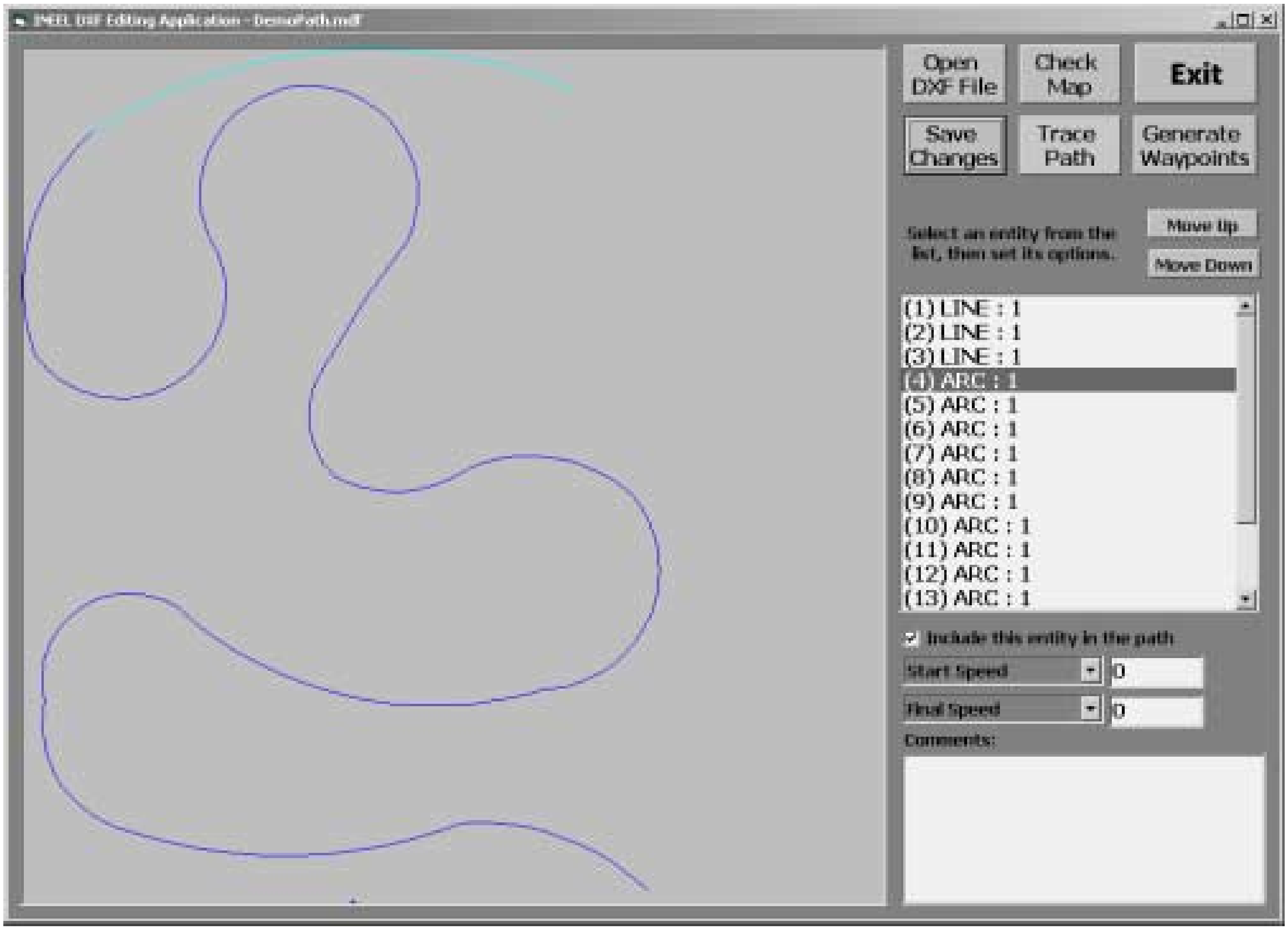

Fig. 1. INEEL DXF Editing Application

- Accurate Positioning and Path Following – The ability of the robotic platform to accurately follow a pre-defined path.
- Overall Automation – The ability to orchestrate path execution together with fully automated data collection from the target, robot and sensor.

II. ROUTE SPECIFICATION

The first major challenge is specifying the desired target route and decomposing this route into a path plan for the robot. To address this issue, the INEEL created IDEA. The INEEL DXF Editing Application (IDEA) is a tool to convert *D*rawing E*x*change *F*ormat (DXF) files into lists of waypoints. This is done in three phases. First, a path is created within any commercial drafting application (e.g., AutoCAD) and saved as a DXF file. Second, IDEA is used to transform this map into a path plan, including the path start and end points; the order of traversal assigned to each segment and the velocity assigned to each vertex. Third, the path plan is decomposed into a list of waypoints that can be sent to the robot. The IDEA tool uses the properties of each entity within the initial DXF file to spatially locate each line or arc and then create a default path based on the initial order of segments found in the DXF.

IDEA can be used to easily manipulate various properties of the initial path. A list of the entities in the path is provided. When a segment is selected in the list, it is highlighted on the image. Figure 1 illustrates a selected arc. System properties are displayed and can be edited, if desired. The order of entities can be changed, either by using the “Move Up” and “Move Down” buttons or by selecting and dragging an entity to its new position. Each segment can be included or excluded from the path by appropriately marking the “Include this entity in the path” checkbox. This allows additional features that are not part of the path to be included in the DXF without the requirement that they be a part of the path as well. Additional input boxes are provided to set the initial speed, the final speed or constant acceleration, and comments for each entity.

Once a DXF has been imported to IDEA and the properties for the entities have been set, the drawing has become a path. The next step is to determine if the path is valid. This can be done at two levels. The “Trace Path” button steps the user through the designated path, highlighting each entity in turn in order to verify order and continuity. The “Check Path” button provides the necessary lower-level check for physical continuity of the path, continuity for the selected speeds for each segment, and the feasibility of the chosen speeds and turn angles in terms of the actual robot’s capabilities. Although the continuity checks for speed and segment order are trivial, the ability to address the physical and operational constraints of the actual robot is a significant challenge.

We strongly believe that all intelligence necessary for the robot to sense and act should reside on the robot itself; therefore, the robot is ultimately responsible for protecting its own safety and that of the environment. The robot will not attempt a turn or speed that is unsafe regardless of the waypoint characteristics which are sent to it. This aspect of the robot’s behavior is referred to as guarded motion. The IDEA software includes a model of the robot’s control algorithm and informs the user of any aspects of the path that will cause the robot’s guarded motion capabilities to activate and subsume the path following algorithm. Once the check has been run, an error report is generated which explains any issues found.

Once the path has been planned out and validated, the user can select the “Generate Waypoints” button which transforms the path into a waypoint file using an algorithm that selects points at intervals along the path based on changes in speed and angle. The resulting waypoint file is a text-based list that includes the coordinates and speed for each waypoint. Once a waypoint file has been generated, it can be used multiple times without having to replan the path.

III. ACCURATE POSITIONING

In order to obtain repeatability and precision, there must be some means to address the fundamental problem of position accuracy. Fully autonomous mobile robots have not entered our homes, factory floors and laboratories *en masse* because such systems have consistently failed to demonstrate the reliability necessary for continuous operation in complex environments. At least one of the primary obstacles to achieving this reliability is position accuracy. After all, mobile robots cannot operate effectively if they do not know where they are. The optimal solution is to have absolute position information. Outdoors, use of the global positioning system (GPS) has provided mobile robots with a crutch that makes it much easier for robots to keep track of where they are. Indoors, however, GPS is not available.

For some tasks, absolute positioning can be achieved by various instrumented solutions such as visual, laser-based tracking systems or radio frequency positioning systems that triangulate distance based on beacons placed in the environment. Each of these systems is costly to implement; in fact, the cost for purchasing and installing such a positioning system is often more than the total cost of the robot itself. Moreover, the utility of vsual or laser tracking systems is limited by occlusions within the environment. RF beacons are only appropriate for environments where the beacons can be fixed in a static, known location. The physical properties of the remote sensing laboratory are constantly changing. In fact, walls are often shifted within the building to model different operational environments. For these reasons, absolute positioning was not deemed a feasible solution. In order to create a robotic solution that could operate in a changing environment, we began to investigate relative positioning methods whereby the robot keeps track of where it is without any external input.

Vehicles that do not utilize an absolute positioning system are vulnerable to odometry drift. The greater distance they travel, the larger the distance error becomes and the more they lose track of where they are in the environment. Even minute errors associated with the use of the industry’s best wheel encoders and inertial sensors can produce unacceptable position error once the robot has moved a significant distance. To address this challenge, the INEEL investigated new localization methods that use sampling of range readings from a scanning laser and ultrasonic sensors to reason probabilistically about where the robot is within its own internal model of the world. A version of Markov localization methods has been incorporated into the INEEL’s robot control system and applied to provide sufficient position accuracy for repeatable, accurate path following within the RSL – Sensor Lab facility.

The robot localization problem is divided into two sub-tasks: global position estimation and local position tracking. Global position estimation is the ability to determine the robot's position in an a priori or previously learned map, given no other information than that the robot is somewhere on the map. Once a robot has been localized in the map, local tracking is the problem of keeping track of that position over time. While existing approaches to position tracking are able to estimate a robot's position efficiently and accurately, they typically fail to globally localize a robot from scratch or to recover from localization failure. Global localization methods are less accurate and often require substantially more computational power. In our project the representation of the robot's state space is based on Monte Carlo sampling [1]. The Monte Carlo technique inherits the benefits of Markovian probability grid approaches for position estimation [2]. This provides an extremely efficient technique for mobile robot localization.

Monte Carlo methods were introduced in 1970 [3] and have recently been applied in the fields of target tracking, computer vision, and robot localization. [1,5] The Monte Carlo Localization (MCL) method is an approach for representing uncertainty in mobile robot localization: instead of describing the state space by a probability density function, it is represented by maintaining a set of samples that are randomly drawn. As the robot moves, an adaptive sampling algorithm [4] determines the number of samples to calculate the probabilistic position distribution. As a result, the MCL method is continuously arbitrating between positional accuracy and computational efficiency. Thus the robot will use many samples to globally locate and fewer samples for positional tracking once the robot is globally located. The sample based representation gives MCL several advantages over prior work in the field of robot localization:

- In contrast to existing positional tracking techniques MCL is able to represent multi-modal distributions for global localization. [1,5]
- The computational complexity of MCL is drastically reduced in comparison to other grid-based Markovian implementations resulting in a smaller, faster localization engine. [1,5]
- MCL is more accurate for a given cell size than other grid-based probabilistic localization methods [1,5]

The goal for this project is to obtain position accuracy of +/- 2cm. At the time that this paper is being written, preliminary tests indicate that this goal may well be met. However, further testing is necessary to assess the limitations of this technique.

It is crucial for the robot to know its position within the environment; however, this alone is not sufficient. Although positioning is the first and foremost challenge, we must also address the control problem of precisely guiding the robot along the predetermined path. This challenge is especially difficult at high speeds. Our approach has been to develop a fuzzy logic controller, which relates turn velocity and translation velocity to the natural logarithm of the angle error. This algorithm allows the robot to accurately follow the designated path and minimizes problems such as the oscillation caused by over-adjustment and the inaccuracy caused by failing to slow down sufficiently in order to make a turn.

IV. OVERALL AUTOMATION

Automation for this task includes the ability to orchestrate path execution together with fully automated data collection from the target, robot, and sensor. For sensor characterization, the *target* is placed on top of the robot and meant to reproduce the size and heat signature of a human. The *sensor(s)* to be tested are placed at a known location within the room and continuously feed data into the Data Acquisition System (DAS) system. The ultimate goal for the end users at the DOE's Sensor Lab is to automate the collection of data from the sensors, target and robot within LabVIEW. LabVIEW is a commercial, off-the-shelf application with which the scientists and engineers at the Sensor Lab are already familiar.

In order to accomplish fully automated data collection, we use an ActiveX control that mediates between the DAS, the robot and the INEEL graphical interface. The ActiveX component can be opened from within the DAS application, which in this case is LabVIEW. The ActiveX component then reads the waypoint supplied by IDEA and sends them out over a specialized communication protocol to the robot via a Free Wave 900MHz radio that insures reliable data link connectivity. In turn, the robot must provide information to the DAS including the vehicle's position, speed and data from the target. LabVIEW receives and correlates data from the robot, the target and associated sensors in such a manner that the performance of each sensor can be accurately compared.

The interface shown in Figure 2 is an ActiveX control, which provides an operator with the ability to access the robot and both send and receive events. The control can be used through any software environment that supports ActiveX control. LabVIEW is not the only DAS that can be used. The interface includes methods that can be called to monitor and initiate robot behaviors. Example methods include downloading a set of waypoints or moving the robot to the start of the path. The ActiveX control also provides event notification for confirmation of actions, such as notification when the robot is positioned at the start of the path or when it has completed the path. The control also exposes various properties for the robot and the target that the DAS can access at any time. These properties are updated in the interface as they are received from the robot, usually whenever they change value, but not less than once per second. Properties include information such as the temperature of the target, the robot location, robot health and communication status. To the monitoring program, the robot interface is just another source of data similar to the sensor readings being collected from the test instruments.

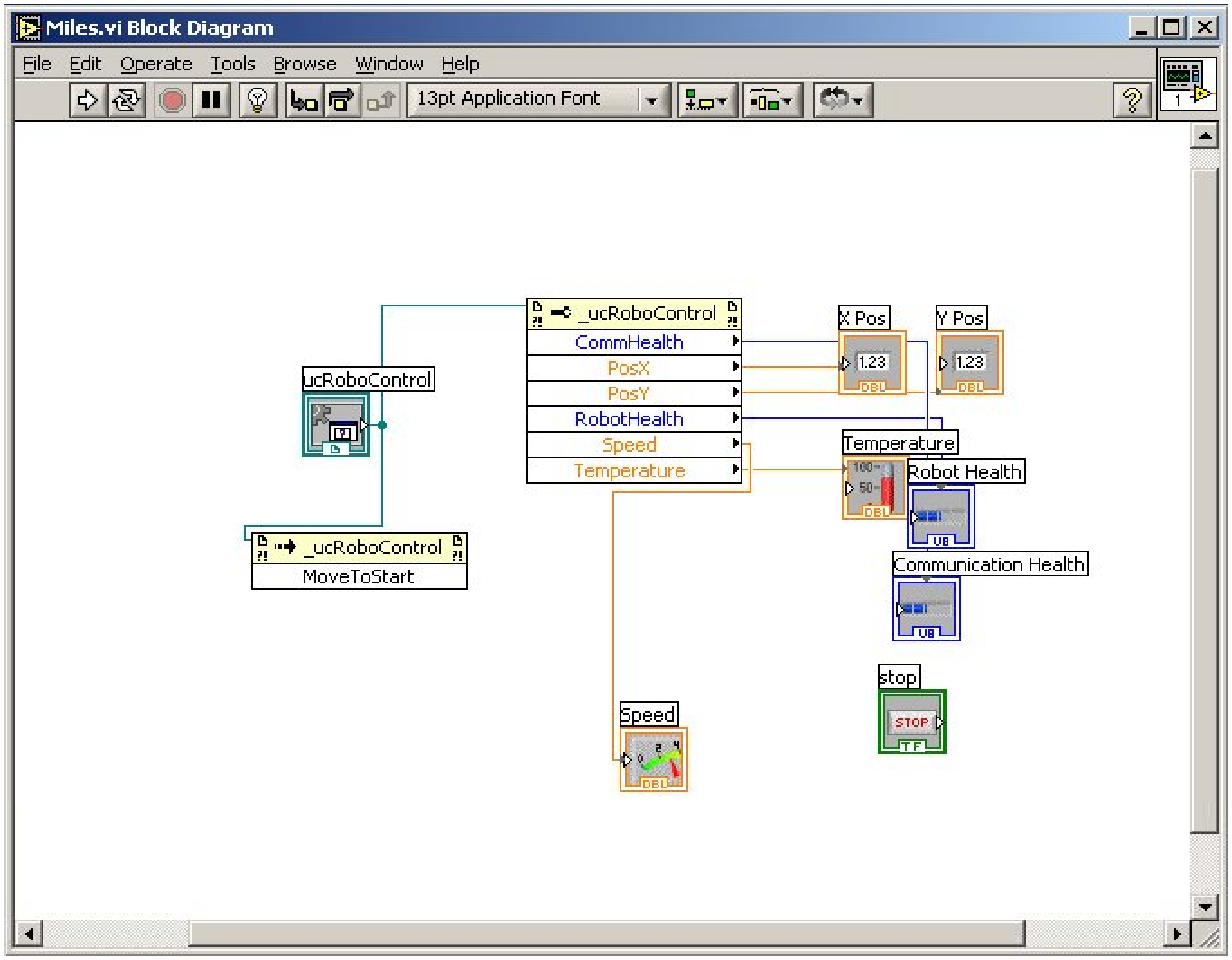

Fig. 2. Selected properties and methods of the ActiveX control from within LabVIEW.

A typical test run would proceed as follows: The researcher develops a path for the robot to follow using a drawing package to generate a DXF file. The path is imported into the waypoint development tool where the various segments are put in order and assigned velocities. The waypoints are then generated and stored in a file. The researcher then starts LabVIEW to run the experiment. In the program the researcher sets the number of times the robot will loop through the given path and starts execution. From within LabVIEW, the ActiveX control is used to uploads the waypoint file to the robot and then signal the robot to move to the start point. Once the robot reaches the start point it signals the DAS and awaits the command to begin. When the order is given, the intelligent controller in the robot executes the path according to the waypoints it has been given and signals the DAS when it completes the path. During the run the robot is continually updating the interface with position, status, and target data that the DAS collects. Once the path is completed the program signals the robot to return to the start point and execute the path again, or return to the home position.

V. CONCLUSIONS

The resulting robotic system, a modified PowerBot, available from ActivMedia, is shown in Figure 3. This system offers dramatic improvements over the current practice of building static tracks on which to move the target, allowing greater automation, efficiency and increased quality of data. Paths can be easily configured and experimental data is automatically collected and sent directly into an analysis software package.

Fig. 3. Robot with target mount

Operational testing within the RSL characterization facility is not yet possible. Informal testing and experimentation indicates that the robot has the ability to accurately follow designated paths at a variety of different speeds. However, it is clear that the accuracy of the path following behavior does diminish as speeds are increased. Once the robot has been installed, we will be able to gather more precise data on the performance of the robot in terms of positioning and path following.

Success in this arena will show that mobile robots can be used across a wide variety of tasks where position accuracy is important. In addition, all capabilities developed for this project have also been ported to other platforms including those shown in Figure 4. In fact, the INEEL intelligent control system can be used for a wide variety of other indoor tasks where the ability to accurately traverse an indoor area is needed.

Fig. 4. Control system has been ported to 6 platforms

ACKNOWLEDGMENTS

This work has been funded by the DOE Remote Sensing Laboratory.

REFERENCES

1. D. FOX, W. BURGARD, F. DELLAERT, and S. THRUN, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *In Proc. of the Sixteenth National Conference on Artificial Intelligence (AAAI'99).*, July, 1999.

2. W. BURGARD, D. FOX, D. HENNIG, and T. SCHMIDT. “Estimating the absolute position of a mobile robot using position probability grids.” In *Proc. of the Thirteenth National Conference on Artificial Intelligence*, pages 896-901, 1996.

3. J. HANDSCHIN. 1970. “Monte Carlo Techniques for prediction and filtering of non-linear stochastic processes.” Automatica6.

4. D. KOLLER, and R. FRATKINA, 1998. “Using learning for approximation in stochastic processes.” Proc. Of ICML-98.

5. F. DELLAERT, D. FOX, W. BURGARD, and S. THRUN, "Monte Carlo Localization for Mobile Robots," IEEE International Conference on Robotics and Automation (ICRA99), May, 1999.